

# Grid-Based Belief Propagation for Cooperative Localization

Panagiotis-Agis Oikonomou-Filandras, Kai-Kit Wong, and Yangyang Zhang

**Abstract**—We present a novel parametric message representation for belief propagation (BP) that provides a novel grid-based way to address the cooperative localization problem in wireless networks. The proposed Grid-BP approach allows faster calculations than non-parametric representations and works well with existing grid-based coordinate systems, e.g., NATO’s military grid reference system (MGRS). This overcomes the hidden challenge inherent in all distributed localization algorithms that require a universally known global reference system (GCS), even though every node localizes using arbitrary local coordinate systems (LCSs) for a reference. Simulation results demonstrate that Grid-BP achieves similar accuracy at much reduced complexity when compared to common techniques that assume *ideal* reference.

**Index Terms**—Belief propagation, cooperative localization.

## I. INTRODUCTION

The unprecedented adoption of mobile handhelds has created a host of new services that require accurate localization, even in GPS-denied environments. How to accurately localize without satellite positioning has been an active research topic. Various cooperative localization methods have been developed suitable to high noise scenarios, e.g., indoors [1, 2].

### A. The Forgotten Challenge

The motivation behind this work is the assumption in the literature of distributed cooperative localization that requires a universally known *global coordinate system (GCS)*. Even in anchor-free algorithms, such as those in [3–5], where nodes localize to a relative *local coordinate system (LCS)* only, the anchors with shared GCS knowledge are required to transform the local coordinates to global ones. Otherwise, nodes would have no idea the whereabouts of the global origin. In the case of distributed ad-hoc networks, where the anchors are simply nodes with a good estimate of their GPS coordinates, achieving a shared GCS between them is non-trivial. Presumably, in this case, the anchors would have to communicate with each other to agree on a GCS with a common origin and pass that information to the rest of the network nodes. In addition, up-to-date information between nodes should be maintained, inducing an increased communication overhead.

The solution we suggest is to use directly a GCS for all calculations, therefore eliminating any need for LCS and all of the described issues. The first obvious choice of a GCS would be to use GPS as a common GCS to all nodes. Unfortunately using GPS coordinates in message passing operations would easily make calculations underflow due to the small distances inherent in indoors localization, and would require scaling and normalization at each node. This means that every node would

have to convert incoming messages to a LCS, suitable for message passing operations, and then convert them back to the GCS for transmission, increasing the computational cost of the algorithm. Instead, we propose a grid-based GCS that can be used directly to conduct message passing operations and at the same time hugely decrease the computational cost. In this manner we both remove any requirement for the networks to consent on a GCS and also achieve very low complexity.

### B. Our Contributions

This correspondence proposes a novel scheme that uses a GCS system suitable for message passing algorithms in cooperative localization. As a real-life example, we use the NATO’s military grid reference system (MGRS), cf. [6].<sup>1</sup> The approach no longer requires GCS coordination between anchors. Also, it has inspired the use of parametric representations using multinomial probability mass functions (pmfs), allowing for a fast robust and accurate cooperative localization algorithm that elegantly resolves the GCS knowledge requirement. In summary, we have made the following contributions:

- We propose a grid-based GCS solution, i.e., map GPS coordinates to unique grid identifiers, solving the common reference issue in all distributed localization techniques.
- Parametric approximations to the pmfs are proposed to overcome the computational bottleneck of non-parametric belief propagation (BP) used in cooperative localization.
- Simulation results illustrate that the proposed grid-based BP method, which is referred to as Grid-BP, provides similar accuracy with low computational cost when compared to common techniques with *ideal* reference.

## II. PROBLEM FORMULATION

We consider a network of nodes in a 2D environment which consists of  $N$  agents and  $M$  anchors, where  $M \geq 4$  and  $N \gg M$ . Let the space be subdivided into a square grid where each square “bucket” has a unique identifier, namely an ID. Then let  $\mathbf{X} = [X_1, \dots, X_i, \dots, X_{N+M}]$  be the locations of all nodes, with  $X_i$  representing the unique identifier of node  $i$  and  $X_i \in \{x_1, x_2, \dots, x_k\}$ , where  $k$  iterates over all possible IDs. Also, let  $\mathbf{Z}$  denote the coordinates of all nodes, with  $\mathbf{Z}_i$  representing the coordinates of node  $i$ , and the domain of  $\mathbf{Z}_i$  is  $\mathbb{R}^2$ . The nodes communicate wirelessly and it is assumed that the maximum communication range for each node is  $R_{\max}$ . Time is slotted and time slots are denoted by the time index superscript  $(t)$  for  $t = 1, 2, \dots, \infty$ . We represent the problem as a joint pmf. Let  $p^{(t)}(X_i)$  be the pmf, i.e., the belief that node  $i$  has about its location at time  $t$ . We model  $p^{(t)}(X_i)$

This work is supported under EPSRC grant EP/H011536/1.

P. Oikonomou-Filandras and K. Wong are with the Department of Electronic and Electrical Engineering, University College London, WC1E 7JE, UK. Y. Zhang is with Kuang-Chi Institute of Advanced Technology, China.

<sup>1</sup>Even though this letter uses the NATO’s MGRS coordinate system, any grid-based coordinate system can be used with trivial changes.

as a multinomial distribution with parameters  $\theta_k$ , where  $\theta_k$  is the probability of node  $i$  being in ID  $x_k$  and  $\sum_k \theta_k = 1$ . In addition, let the set of all nodes  $j$  within range of node  $i$  be denoted as the neighbourhood  $\mathcal{N}_i$ . Initially, the belief for the agents can be a non-informative uniform pmf over the grid, while the anchors' pmfs are focused in the IDs close to the real position, e.g., within 10m. Anchors can obtain their IDs either by directly mapping them from satellite data, cf. [6]. Node  $i$  receiving a message from node  $j$  at time slot  $t$  can derive, using time-of-arrival (ToA) measurements,<sup>2</sup> a noisy estimate  $r_{j \rightarrow i}^{(t)}$  of the distance between them. For convenience, we assume  $r_{j \rightarrow i}^{(t)} = r_{i \rightarrow j}^{(t)} = r_{ji}^{(t)}$ .

Thus, as in [7] for ToA distance measurements, we define the random variable  $R_{ji}$  with its value  $r_{ji}$  modelled as

$$r_{ji} = \|z_i - z_j\| + \eta_{ji}, \quad (1)$$

where  $\eta_{ji}$  is a Gaussian noise with variance  $\sigma_{ji}^2 = K_e \|z_i - z_j\|^{\beta_{ji}}$  in which  $K_e$  is a proportionality constant capturing the combined physical layer and receiver effect, and  $\beta_{ji}$  denotes the path loss exponent. In the case of line-of-sight (LoS),  $\eta_{ji}$  is assumed zero mean, and  $\beta_{ji} = 2$ , i.e.,  $\eta_{ji} \sim \mathcal{N}(0, \sigma_{ji}^2)$ . In this work we assume only LoS, but it would be easy to extend the algorithm with NLoS mitigation, by using e.g. [8, 9].

We define the likelihood of node  $i$  and node  $j$  measuring distance  $R_{ji} = r_{ji}$  between them at time  $t$ , given  $X_i, X_j$  as

$$p^{(t)}(R_{ji} = r_{ji} | X_i, X_j) \propto \exp \left( - \left( \frac{r_{ji} - \|C_i - C_j\|_2}{h} \right)^2 \right) \quad (2)$$

where  $h$  controls steepness,  $C_i$  and  $C_j$  are the coordinates of the centers of the grids' squares  $X_i$  and  $X_j$ , respectively. Therefore, our objective is to find the maximum a posteriori (MAP), i.e., the values that maximize  $p(\mathbf{X}|\mathbf{R})$  given distance measurements  $\mathbf{R} = [R_{ji}]$ . For a specific node  $i$ , we have

$$\hat{X}_i = \arg \max_{X_i} p^{(t)}(X_i | \mathbf{R}_i). \quad (3)$$

Thus,  $p^{(t)}(X_i | \mathbf{R}_i)$  can be evaluated using the Bayes' rule as

$$\begin{aligned} p^{(t)}(X_i | \mathbf{R}_i) &\propto p^{(t)}(X_i) \prod_{j \in \mathcal{N}_i} p^{(t)}(R_{ji} | X_i) \\ &\propto p^{(t)}(X_i) \prod_{j \in \mathcal{N}_i} \int p^{(t)}(R_{ji} | X_i, X_j) p^{(t)}(X_j) dX_j, \end{aligned} \quad (4)$$

in which the sign “ $\propto$ ” means “is proportional to”, and normalization should be done to obtain the pmf.

### III. THE PROPOSED GRID-BP ALGORITHM

Cooperative localization can be viewed as running inference on a probabilistic graphical model (PGM) [2], where messages are representations or probability functions. Here, the proposed grid-based localization algorithm will be presented. First, the use of a cluster graph and BP will be motivated, and the used PGM will be analysed. Then efficient approximation for the marginalization and the product operation will be given.

<sup>2</sup>The assumption of using ToA is not restrictive on the proposed algorithm because it can easily be used with other measurement models.

#### A. Belief Message Passing

The network can be modelled as a cluster graph and loopy belief message passing algorithm can be used. We adopt a Bethe cluster graph [10]. The lower factors are composed of univariate potentials  $\psi(X_i)$ . The upper region is composed of factors equal to  $\psi(X_i, X_j, R_{ji})$ , e.g., see Fig. 1.

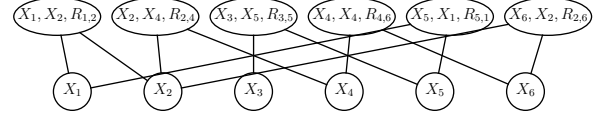


Fig. 1. The cluster graph. Lower row factors denote the node position beliefs. Upper row factors denote the ranging interactions between the nodes.

The lower factors are set to the initial beliefs for the given time slot ( $t$ ), and the upper factors to the corresponding cpmfs:

$$\psi(X_i) = p^{(t)}(X_i), \quad (5)$$

$$\psi(X_i, X_j, R_{ji} = r_{ji}^{(t)}) = p^{(t)}(R_{ji} = r_{ji}^{(t)} | X_i, X_j). \quad (6)$$

Messages are then passed between nodes for multiple iterations until the node beliefs have converged. The message from node  $j$  to node  $i$ , at BP iteration  $(s+1)$  is calculated by

$$\mu_{j \rightarrow i}^{(s+1)}(X_i) = \int \psi(X_i, X_j, R_{ji} = r_{ji}^{(t)}) \frac{b_{j \rightarrow i}^{(s)}(X_j)}{\mu_{i \rightarrow j}^{(s)}(X_j)} dX_j, \quad (7)$$

where intuitively, a message (7) is the belief that node  $j$  has about the location of node  $i$  and  $r_{ji}^{(t)}$  is the observed value of the distance between the nodes, at time slot  $t$ .

Then the belief of node  $i$  is updated as

$$b_i^{(s+1)}(X_i) = \lambda \psi(X_i) \prod_{k \in \mathcal{N}_i} \mu_{k \rightarrow i}^{(s+1)}(X_i) + (1-\lambda) b_i^{(s)}(X_i), \quad (8)$$

where  $\lambda$  is a dampening factor used to facilitate convergence. BP continues until convergence, or if convergence is not guaranteed,  $s$  reaches a maximum number of iterations  $I_{\max}$ . Then the beliefs, representing approximations to the true marginals, for each node are found by (8), i.e.,  $p^{(t+1)}(X_i) = b_i^{(s+1)}(X_i)$ . The proposed Grid-BP is given as Algorithm 1. Each node needs to perform a marginalization operation (7), and a product operation (8). Approximations are required for both complex operations. In Grid-BP, we take advantage of the multinomial parametric form which we discuss next.

#### B. Marginalization Operation (7)

The calculation of (7) gives the belief node  $j$  has about node  $i$ . To understand this, let us assume the case where all energy in  $b_j^{(s)}(X_j)$  is concentrated at a single ID  $x_j$ , then node  $j$  would believe that node  $i$  is located in one of the IDs that approximate a “circle” with centre  $x_j$  and radius  $r_{ji}^{(t)}$ . Hence, to get  $\mu_{j \rightarrow i}^{(s+1)}(X_i)$ , first we draw  $L$  particles from  $x_j^{(l)} \sim b_j^{(s)}(X_j)$ . Then we draw  $L$  samples from  $\phi^{(l)} \sim \mathcal{U}(0, 2\pi)$  and  $L$  samples from  $\hat{r}_{ji}^{(l)} \sim \mathcal{N}(r_{ji}^{(t)}, h)$ . The Gibbs sampling algorithm is provided as Algorithm 2. We repeat Algorithm 2 for all incoming messages and we will get  $\{x_j^{(l)}, \hat{r}_{ji}^{(l)}, \phi^{(l)}\}_{l=1, j=1}^{L, |\mathcal{N}_i|}$ .

---

**Algorithm 1** Grid-BP

---

```

1: Initialize beliefs  $p^{(0)}(X_i) \forall i \in \text{Nodes}$ 
2: for  $t = 0$  to  $T$  do
3:   for all  $i \in \text{Nodes}$  do
4:     Broadcast current belief  $p^{(t)}(X_i)$ 
5:     for all  $j \in \mathcal{N}_i$  do
6:       Collect distance estimates  $r_{ji}^{(t)}$ 
7:     end for
8:   end for
9:   Initialize  $\psi(X_i) = p^{(t)}(X_i)$ 
10:  Initialize  $\psi(X_i, X_j, R_{ij}) = p^{(t)}(R_{ij} = r_{ji}^{(t)} \mid X_i, X_j)$ 
11:  repeat
12:    for all  $i \in \text{Nodes}$  do
13:      for all  $j \in \mathcal{N}_i$  do
14:        Receive  $b_j^{(s)}(X_j)$ 
15:        Calculate  $\mu_{j \rightarrow i}^{(s+1)}(X_i)$ , using (7) using Gibbs sampling (i.e., Algorithm 2)
16:      end for
17:      Calculate  $b_i^{(s+1)}(X_i)$ , using (8).
18:      Check for convergence
19:      Send  $b_i^{(s+1)}(X_i)$ 
20:    end for
21:  until convergence or  $s$  reaches  $I_{\max}$ 
22:  Update belief  $p^{(t+1)}(X_i)$ , using (8).
23: end for

```

---



---

**Algorithm 2** Grid Gibbs Sampling

---

```

1: Set  $\mathcal{D}_{X_i}$  to empty
2: for all  $j \in \mathcal{N}_i$  do
3:   Sample  $x_j^{(l)} \sim \mu_{j \rightarrow i}(X_j)$  which is a multinomial pdf
4:   Sample  $\phi^{(l)} \sim \mathcal{U}[0, 2\pi]$ 
5:   Sample  $\hat{r}_{ji}^{(l)} \sim \mathcal{N}(r_{ji}^{(t)}, h)$ 
6:    $x_i^{(l)} = \text{MAP-DMtoID}(x_j^{(l)}, \hat{r}_{ji}^{(l)}, \phi^{(l)})$  which maps the distance metric to IDs
7:   Add  $\{x_i^{(l)}\}_{l=1}^L$  to  $\mathcal{D}_{X_i}$ 
8: end for
9: return  $\mathcal{D}_{X_i}$ 

```

---

It is important to note that in order to combine the distance metric with the sampled IDs  $\{x_j^{(l)}\}$  and get the set  $\mathcal{D}_i = \{x_i^{(l)}\}$ , we use a mapping function which we define as

$$x_i^{(l)} = \text{MAP-DMtoID}(x_j^{(l)}, \hat{r}_{ji}^{(l)}, \phi^{(l)}) \quad (9)$$

Intuitively we do this by counting for each sampled ID  $x_j^{(l)}$  the number of IDs to the east and to the north, node  $i$  will be, given the measured distance samples  $\hat{r}_{ji}^{(l)}$  normalized by  $D$  as

$$\begin{bmatrix} d_H \\ d_V \end{bmatrix} = \text{int} \left( \frac{\hat{r}_{ji}^{(l)}}{D} \begin{bmatrix} \cos(\phi^{(l)}) \\ \sin(\phi^{(l)}) \end{bmatrix} \right). \quad (10)$$

Then we map the displacement  $d_H, d_V$  to a new ID and return it as  $x_i^{(l)}$ . The set  $\mathcal{D}_{X_i}$  of all samples obtained from all incoming messages is used to find (8). The distance to ID mapping function is given as Algorithm 3. In the case of MGRS IDs the horizontal and vertical mappings are done by

adding  $d_H, d_V$  to the easting and northing components of the ID of  $x_j^{(l)}$ , to be discussed in Section IV. There is no need to do any reverse mapping as we directly get the new IDs.

---

**Algorithm 3** MAP-DMtoID

---

```

1: Calculate horizontal and vertical steps using (10)
2: Map horizontal ID  $x_j^{(l)} \rightarrow b_j^{(l)}$ 
3:  $b_h^{(l)} = b_j^{(l)} + d_H$ 
4: Inverse horizontal mapping  $b_h^{(l)} \rightarrow x_h^{(l)}$ 
5: Map vertical ID  $x_h^{(l)} \rightarrow b_v^{(l)}$ 
6:  $b_i^{(l)} = b_v^{(l)} + d_V$ 
7: Inverse vertical mapping  $b_i^{(l)} \rightarrow x_i^{(l)}$ 
8: return  $x_i^{(l)}$ 

```

---

**C. Product Operation (8)**

To obtain (8), we will first convert the samples calculated for each message  $\mu_{j \rightarrow i}^{(s+1)}(X_i)$ , (7), to parametric multinomial pmfs and then calculate their product. We assume that the parameters of each multinomial are random variables  $\Theta_i$  with a uniform Dirichlet prior with parameters  $\alpha = [\alpha_1 \cdots \alpha_K]$ , where  $K$  denotes the number of unique IDs in the pmf, i.e.  $\mu_{j \rightarrow i}^{(s+1)}(X_i) \approx p_{j \rightarrow i}^{(s+1)}(X_i \mid \Theta_i; \alpha)$ . Firstly we use the samples from each incoming message as observations and get the MAP estimate  $\hat{\theta}_i = [\hat{\theta}_{i,1} \cdots \hat{\theta}_{i,K}]$  of the parameters  $\Theta_i$  of each  $p_{j \rightarrow i}^{(s+1)}(X_i \mid \Theta_i; \alpha)$ . We also assume that all incoming messages have the same prior distribution. This allows us to efficiently create parametric forms of the incoming messages as multinomial pmfs. These parameters are calculated as

$$\hat{\theta}_i = \mathbb{E}[P(\Theta_i \mid \mathcal{D}_{X_i})] = \mathbb{E}[p(\mathcal{D}_{X_i} \mid \Theta_i)p(\Theta_i)], \quad (11)$$

which gives

$$\hat{\theta}_{i,k} = \frac{M_k + |\mathcal{N}_i| \alpha_k}{|\mathcal{N}_i| \sum_k (M_k + \alpha_k)}, \quad (12)$$

where  $M_k$  is number of particles  $x_k$  in  $\mathcal{D}_{X_i}$ . The algorithm is presented in Algorithm 4. For clarity, the quantities of each ID in the samples are shown as being found by a count function but in practice it can be done during the Gibbs sampling step allowing for a more efficient algorithm. Finally, with the parametric forms of each (7) in hand, we can easily calculate (8) as the dot product of the  $\theta_i$  parameters of all incoming messages and the node's own belief.

---

**Algorithm 4** MAP Parameter Estimation

---

```

1: Let  $|X_i|$  be the number of unique IDs in  $p(X_i \mid \Theta_i)$ 
2: Calculate  $M_k = \text{count}(x_k, \mathcal{D}_{X_i}) \forall k \in |X_i|$ 
3: for all  $k \in |X_i|$  do
4:   Calculate  $\hat{\theta}_{i,k}$ , with (12)
5: end for
6: return  $\hat{\theta}_i$ 

```

---

#### D. Message Filtering

As it makes no sense to keep all the possible IDs on the planet wide GCS, we consider that each node constructs pmfs with the IDs only within a specific range  $R_{\text{grid}}$ , e.g., within 100m, of the IDs it receives in the first iteration. To reduce the IDs further, we propose a simple filter to only keep the most probable IDs summing up to an energy threshold. Simulation results (not included in this paper) suggest that by keeping  $\sim 80\%$  of the total energy of the pmf, the size of the messages is decreased by  $\sim 90\%$  with no increase in localization error. Thus, assuming that each message covers a  $100 \times 100\text{m}^2$  grid, the total number of IDs used without the filter would be  $10^4$ . After the filter, only  $\sim 100$  IDs will be transmitted.

#### E. Complexity

Most message passing cooperative localization algorithms tend to use a variant of Gibbs sampling to calculate (7) with similar computational costs [11]. Hence, the complexity cost tends to be bounded by the product operation (8) cost. As Grid-BP multiplies parametric form multinomials, the operation is bounded by  $\bar{K}(|\mathcal{N}_i| + 1)$ , where  $\bar{K}$  is the average number of IDs used in the calculations. The computational cost for the product operation of the compared algorithms is summarized in Table I, where  $L$  is the number of particles and  $|\mathcal{N}_i|$  is the number of messages involved and  $I_{\text{Hybrid-BP}}$  is the number of iterations the product algorithm in [12] is run.

TABLE I  
COMPARISON OF COMPLEXITY COSTS OF (8).<sup>§</sup>

Approach	Algorithm	Complexity
Non-parametric	NBP	$L_{\text{NBP}}^2( \mathcal{N}_i  + 1)$
Non-parametric	HEVA-BP	$L_{\text{HEVA}}^2( \mathcal{N}_i  + 1)$
Parametric	Grid-BP	$\bar{K}( \mathcal{N}_i  + 1)$
Parametric	Hybrid-BP	$I_{\text{Hybrid-BP}} L_{\text{NBP}}  \mathcal{N}_i  ( \mathcal{N}_i  + 1)$

<sup>§</sup>Note that  $\bar{K} \simeq L_{\text{HEVA}} \ll L_{\text{NBP}}$ , and typically  $I_{\text{Hybrid-BP}} \approx 100$ .

#### IV. REVIEW OF MGRS

In this correspondence, as a grid based system, we employ MGRS, which is the geo-coordinate standard used by NATO military for locating points on the planet [6] and a combination of the universal transverse mercator (UTM) grid system and the universal polar stereographic (UPS) grid system, with a different labelling convention. It is essentially a global mesh grid that assigns a unique ID to each grid square. An example ID is 10QCG12345678, where the first part “10Q” is called the grid zone designator (GZD), the second part “CG” is the 100,000-meter-square identifier, and finally the last numerical part gives the easting (first half digits) and northing (second half digits) inside the square identifier. Every two digits used (for a minimum of 2 and a maximum of 10) increase the resolution by a factor of 10m, down to a resolution of  $1\text{m}^2$  grid squares. Map coordinates are read from west to east first (easting), then from south to north (northing), i.e., left-right, down-up. In cases where the part of the ID is common to all neighbouring nodes, the common part can be dropped and only the rest need to be transmitted or used. For details on the specifics of MGRS, readers are referred to [6].

#### V. SIMULATION RESULTS

To assess the performance of Grid-BP, we conducted 300 Monte-Carlo simulations and the root-mean-square (RMS) localization error was calculated for various noise levels.

In each simulation 100 nodes with 20 anchors are placed randomly in a  $100\text{m} \times 100\text{m}$  area and the communication range is limited to 12m and  $|\mathcal{N}_i|_{\text{avg}} = 4.03$ . Anchor locations are modelled as multivariate Gaussian pdfs with an identity variance matrix. The grid resolution for Grid-BP is  $D = 1\text{m}$ . We compare Grid-BP with HEVA-BP [8], a computationally cheaper variation of non-parametric BP (NBP) [13]. We also compare it with a parametric belief propagation algorithm, namely Hybrid-BP [12].<sup>3</sup> In addition, the maximum number of message passing iterations is set to be  $I_{\text{max}} = 15$ . The experiments were run for different noise levels with the noise factor used ranging between  $K_e = [0.0 - 0.3]$  and the number of particles being 200. Furthermore, to showcase the increased complexity of using GPS coordinates as a common GCS, a variant of HEVA that uses GPS was also provided. In HEVA-GPS, messages contain GPS coordinates that every node converts to an LCS before calculating (7) and (8). Afterwards the updated beliefs are converted back to GPS coordinates and transmitted.

In Fig. 2, the RMS localization error of all the algorithms as the noise coefficient increases is shown. We see that HEVA-BP and HEVA-BP-GPS have a better accuracy than both Grid-BP and Hybrid-BP. Grid-BP has a slightly higher RMS error.

In Fig. 3, the average simulation time against the number of particles is shown. The improvement in computational cost by Grid-BP is observed, especially as the number of particles increases. It should also be noted that for higher particle numbers the cost of GPS scaling becomes almost insignificant compared to the cost of message passing equations.

Note that both algorithms, HEVA and Hybrid-BP have the strong assumption of sharing knowledge of the GCS origin, while Grid-BP and HEVA-GPS do not (the realistic scenario). Even though HEVA-BP-GPS performs as good as HEVA-BP, there is an increase in computational cost with HEVA-BP due to the mapping of the GPS coordinates to a local Cartesian reference frame (as can be observed in the mean simulation time in Fig. 3). As the number of particles gets higher, the relative computational efficiency of Grid-BP can also be seen. Note that all the simulations shown were run on an Intel i7 2.6GHz, using Python for scientific computing [14].

#### VI. CONCLUSION

This correspondence has proposed a novel parametric BP algorithm for cooperative localization that uses a grid-based system. The resulting Grid-BP algorithm combines a grid-based GCS that alleviates the hidden issue of requiring shared reference knowledge, and a parametric representation which allows quick and efficient inference. Simulation results showed that Grid-BP’s performance is significantly better than other BP algorithms that rely on a known GCS. Grid-BP is also

<sup>3</sup>For Hybrid-BP [12], we did not use information given from satellites.

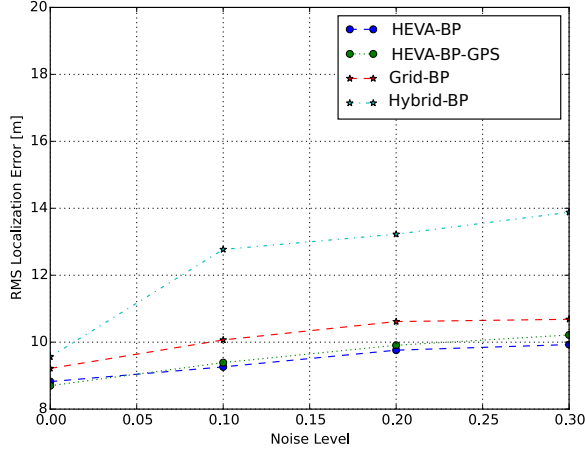


Fig. 2. The rms error versus the amount of range estimation noise  $K_e$  when the average node connectivity is 4.03

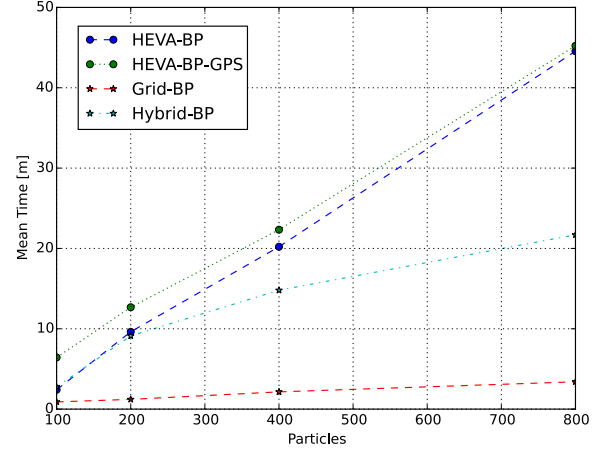


Fig. 3. The average time [m] versus the amount of particles used.

easy to be extended for mobile applications and add the non-LoS mitigation filter proposed in [8], making it a versatile and reliable choice in both military and civilian applications.

## REFERENCES

- [1] N. Patwari *et al.*, "Locating the nodes: Cooperative localization in wireless sensor networks," *IEEE Signal Process. Mag.*, vol. 22, no. 4, pp. 54–69, Jul. 2005.
- [2] H. Wymeersch, J. Lien, and M. Win, "Cooperative localization in wireless networks," *Proc. IEEE*, vol. 97, no. 2, pp. 427–450, Feb 2009.
- [3] A. Ihler *et al.*, "Nonparametric belief propagation for sensor self-calibration," in *Proc. IEEE Int. Conf. Acoustics, Speech, and Signal Process.*, vol. 3, May 2004, pp. iii–861–4 vol.3.
- [4] N. B. Priyantha *et al.*, "Anchor-free distributed localization in sensor networks," in *Proc. Int. Conf. Embedded Networked Sensor Systems*, ACM, 2003, pp. 340–341.
- [5] Y. Shang and W. Ruml, "Improved mds-based localization," in *Proc. IEEE INFOCOM*, vol. 4, 2004, pp. 2640–2651.
- [6] "Military Map Reading 201", [Online]. Available: <http://http://earth-info.nga.mil/GandG/coordsys/mmr201.pdf>
- [7] R. Buehrer, T. Jia, and B. Thompson, "Cooperative indoor position location using the parallel projection method," in *Proc. Int. Conf. Indoor Positioning Indoor Navigation*, Sep. 2010, pp. 1–10.
- [8] P.-A. Oikonomou-Filandras and K. K. Wong, "Hybrid non-parametric belief propagation for localization in wireless networks," in *Proc. Sensor Signal Process. for Defence*, Sep. 2011, pp. 1–5.
- [9] Yin, Feng, *et al.* "Cooperative localization in WSNs using Gaussian mixture modeling: Distributed ECM algorithms," *IEEE Trans. Signal Process.*, vol. 63, no. 6, pp. 1448–1463, Mar 2015.
- [10] D. Koller and N. Friedman, *Probabilistic graphical models: Principles and techniques*, Cambridge, MA : MIT Press, 2009.
- [11] J. Lien *et al.*, "A comparison of parametric and sample-based message representation in cooperative localization," *Int. J. of Navigation and Observation*, vol. 2012, p. 10, 2012.
- [12] M. Caceres *et al.*, "Hybrid cooperative positioning based on distributed belief propagation," *IEEE J. Sel. Areas Commun.*, vol. 29, no. 10, pp. 1948–1958, Dec. 2011.
- [13] A. Ihler *et al.*, "Nonparametric belief propagation for self-localization of sensor networks," *IEEE J. Sel. Areas Commun.*, vol. 23, no. 4, pp. 809–819, Apr. 2005.
- [14] T. E. Oliphant, "Python for scientific computing," *Computing in Science & Engineering*, vol. 9, no. 3, pp. 10–20, 2007.